IN THE CLAIMS:

Please amend claim 10 as follows and cancel claims 12-36.

1.   (Original)   A system to synchronize object management systems having a plurality of object management system components, comprising:

a distributed reader and writer's lock for each of the plurality of object management system components that communicates over a language interface and controls access to information shared by a corresponding object management system component within multiple object management systems;

a module that creates the distributed reader and writer's lock and serves as an agent for the object management system component; and

a list controller, which maintains a communications list and is adapted for communication with a communications controller, to which the distributed reader and writer's lock offloads management of a stub interface; wherein the distributed reader and writer's lock functions to:

request a local read lock and release a read lock;

request a local write lock and release a local write lock; and

request a remote write lock and release a remote write lock.

2.   (Original)   The system of claim 1, wherein each distributed reader and writer's lock communicates with corresponding locks on other object management systems through an ix_ring object that serves as a ring buffer.

3.   (Original)   The system of claim 1, wherein the distributed reader and writer's lock provides two callback functions while registering as a client of the list controller.

4.   (Original)   The system of claim 3, wherein one callback function is the creation and initialization of an ix_base_t object as a stub interface to a new object management system being connected to the system.

5.   (Original)   The system of claim 4, wherein the language interface performs the initialization of the ix_base_t object generated stub initialization function.

6.   (Original)   The system of claim 3, wherein one callback function is the clean up and destruction of an ix_base_t object when an object management system is disconnected from the system.

7.   (Original)   The system of claim 6, wherein the language interface performs the clean up of the ix_base_t object generated stub clean up function.

8.   (Original)   The system of claim 1, wherein the module may have only one write lock at a time and several read locks.

9.   (Original)   The system of claim 8, wherein the write lock is granted to the module upon a release of all outstanding read locks and a grant and release of all outstanding read lock requests.

10.   (Currently amended)   The system of claim 1, wherein the language interface uses an ix_base_t object to support a skeleton interface, which supports an incoming message, and an ix_base_t object to support [[a]] the stub interface, which supports an outgoing message.

11.   (Original)   The system of claim 1, wherein the list controller provides a function to iterate through the distributed reader and writer's lock ix_base_t objects.

12.   (Canceled)

13.   (Canceled)

14.   (Canceled)

15.   (Canceled)

16.   (Canceled)

17.   (Canceled)

18.   (Canceled)

19.   (Canceled)

20.   (Canceled)

21.   (Canceled)

22.   (Canceled)

23.   (Canceled)

24.   (Canceled)

25.   (Canceled)

26.   (Canceled)

27.   (Canceled)

28.   (Canceled)

29.   (Canceled)

30.   (Canceled)

31.   (Canceled)

32.   (Canceled)

33.   (Canceled)

34.   (Canceled)

35.   (Canceled)

36.   (Canceled)